RFC 6518 : Keying and Authentication for Routing Protocols (KARP) Design Guidelines

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 février 2012

Date de publication du RFC : Février 2012

https://www.bortzmeyer.org/6518.html

Dans l'ensemble du travail engagé pour améliorer la sécurité du routage sur l'Internet, un sous-problème important est celui de la gestion des clés. En cryptographie, c'est souvent par une faiblesse dans cette gestion que les systèmes de sécurité sont compromis. Le groupe de travail KARP <http://tools.ietf.org/wg/karp> de l'IETF est occupé à améliorer les protocoles de gestion de clés et son premier RFC, ce RFC 6518 [1], expose les propriétés attendues des futurs protocoles de gestion des clés des routeurs.

Prenons par exemple N routeurs OSPF qui veulent s'authentifier les uns les autres. La technique du RFC 2328 est d'avoir un secret partagé par tous les routeurs du même réseau local. Les messages OSPF sont concaténés avec ce secret, le résultat de la concaténation est ensuite condensé cryptographiquement, ce qui permettra au destinataire de s'assurer que l'émetteur connaissait bien le secret. Ce secret partagé est une clé cryptographique. Qui va la générer ? Comment la distribuer de façon sûre ? Comment la changer facilement et rapidement si le secret est compromis (ou, tout simplement, si un employé quitte l'entreprise) ? Ce genre de questions est la problématique de **gestion de clés**. Dans le cas d'OSPF, actuellement, la quasi-totalité des opérateurs de routeurs fait cela à la main (on se logue sur chaque routeur et on change le secret...) ou, à la rigueur, via un protocole général de configuration des routeurs. Peut-on faire mieux ? C'est en tout cas ce que va essayer de faire le groupe KARP (les deux RFC suivants de KARP ont été le RFC 6862, sur l'analyse des menaces, et le RFC 6863, sur l'analyse spécifique du protocole OSPF).

C'est que les mécanismes actuels ne sont pas satisfaisants. Comme le rappelle la section 1 du RFC, lors d'un colloque en 2006 (cf. RFC 4948), les participants avaient dénoncé la vulnérabilité des routeurs

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6518.txt

aux tentatives de prise de contrôle et appelé à durcir leur sécurité. Quatre axes d'amélioration avaient été identifiés, améliorer la gestion des routeurs (groupe de travail OPSEC <http://tools.ietf.org/wg/opsec>), améliorer les IRR, valider les annonces de route (groupe de travail SIDR <http://tools.ietf.org/wg/sidr>), et enfin sécuriser les communications entre routeurs (groupe de travail KARP <http://tools.ietf.org/wg/karp>), ce qui fait l'objet de ce RFC. Les informations de routage sont échangées via un protocole et la protection de ce protocole se fait par la cryptographie. Qui dit cryptographie dit clés. Lorsqu'un routeur cherche une clé pour protéger un message, où demande-t-il ? Pour l'instant, il n'y a pas de mécanisme standard et KARP va essayer d'en développer un. Le plus courant aujourd'hui est la gestion manuelle des clés (l'opérateur configure les clés, les change à la main - lorsqu'il y pense, les communique via PGP, SCP voire sans aucune sécurité) mais le RFC estime que le futur est dans des mécanismes automatisés de gestion de clés, les **KMP** (pour "*Key Management Protocol*"). Un KMP, par exemple, change automatiquement les clés au bout d'une période pré-définie.

Compte-tenu de la variété des protocoles de routage, et du transport qu'ils utilisent, ce sera un mécanisme abstrait, pas un protocole précis (ce RFC 6518 n'a donc pas d'implémentation, il définit juste un cadre). Plus précisement, KARP va concevoir les interfaces abstraites entre le système de gestion de clés et le protocole de routage, puis, pour chaque protocole de routage, la correspondance entre cette interface abstraite et le protocole réel. Un projet ambitieux.

Maintenant, au boulot. Qu'est-ce qui est déjà fait dans ce RFC ? La section 2 classe les protocoles de routage selon leurs propriétés. L'idée est que les protocoles de routage qui partagent les mêmes propriétés pourront, avec un peu de chance, utiliser les mêmes mécanismes de gestion de clés. Première propriété, le type de message. Certains protocoles sont de type un-vers-un : les messages d'un routeur sont envoyés à un autre routeur. Les `UPDATE` BGP (RFC 4271) fonctionnent ainsi. Mais c'est aussi le cas de LDP (RFC 5036), de BFD (RFC 5880) et même d'OSPF (RFC 2328) dans certaines conditions (liens point -à-point). D'autres protocoles fonctionnent en un-vers-plusieurs. Un routeur diffuse sur le réseau local l'information de routage. C'est le mode de fonctionnement le plus courant d'OSPF et c'est aussi celui de RIP (RFC 2453). Enfin, il y a les protocoles utilisés pour le "*multicast*".

Deuxième propriété pour classer les protocoles de routage, proche de la précédente mais pas identique, le fait que la clé soit par groupe ou par pair. Dans BGP ou LDP, les clés sont individuelles, on a une clé différente par pair. Dans OSPF, la clé est la même pour tous les pairs d'un groupe.

La section 3 liste ensuite les points auxquels il faut penser lorsqu'on envisage un protocole de gestion de clés, un KMP. Le RFC 4107 fournit les bases générales et notre RFC l'adapte aux protocoles de routage. Entre autres, il faudra penser aux paramètres à passer avec le système de gestion de clés, comme la durée de vie pour les clés, l'identificateur de l'association de sécurité (SPI dans IPsec, KeyID dans TCP-AO, etc), l'algorithme de chiffrement et plusieurs autres.

Deux points sont soulignés par le RFC, les questions particulières des clés asymétriques (section 3.1) et le cycle de vie des clés (section 3.2). Les clés asymétriques sont souvent une bonne solution aux problèmes de sécurité : déjà utilisées par les routeurs (lorsqu'ils sont administrés via SSH), générées sur le routeur, elles peuvent n'avoir jamais besoin de le quitter (le RFC ne le dit pas clairement mais on peut même imaginer un petit HSM dans le routeur). Générées aléatoirement, elles ne peuvent pas être devinées comme le sont tant de mots de passe choisis par des humains. Il faut juste faire attention à leur taille, pour limiter les risques des attaques par force brute (RFC 3766). L'algorithme classique pour ces clés est RSA mais les algorithmes à base de courbes elliptiques commencent à se répandre, et permettent d'utiliser des clés plus courtes.

Quant au cycle de vie des clés, notre RFC insiste surtout sur la nécessité d'avoir des remplacements ("*rollover*") de clés qui soient discrets : un remplacement de clés ne devrait pas casser les sessions de routage existantes, car cela imposerait des recalculs lourds des tables de routage, se propageant dans

tout le réseau, et entraînant parfois des perturbations dans la connectivité. (Le RFC ne le cite pas mais la traditionnelle authentification MD5 de BGP - RFC 2385 - n'a pas cette propriété. Changer la clé impose de relancer les sessions BGP. C'est sans doute une des raisons pour lesquelles ces clés ne sont jamais changées, même quand tout le monde les connait.)

Pourquoi, d'ailleurs, faut-il changer les clés de temps en temps ? Il peut y avoir des raisons cryptographiques (progrès de la cryptanalyse, mais le RFC note qu'en pratique, ce sont les cas les plus rares, des problèmes moins prestigieux scientifiquement sont bien plus communs), des raisons liées au personnel (départ d'un ingénieur qui connaissait les clés), des raisons plus urgentes (compromission d'une machine où étaient stockées des clés). Même s'il n'y a aucune raison immédiate de changer, un remplacement des clés de temps en temps peut être nécessaire pour s'assurer qu'un attaquant qui a obtenu une clé et l'utilise discrètement (de manière purement passive), ne puisse pas profiter de son butin éternellement.

Lorsque les procédures de changement de clés sont manuelles, les changements peuvent être en eux-mêmes une source de vulnérabilité (l'erreur est humaine...).

Après ces préliminaires, la section 4 dessine le travail futur. Il y a deux chantiers génériques (indépendants du protocole de routage) importants, le premier étant un pré-requis du second :

— Doter tous les protocoles de routage de mécanismes leur permettant l'agilité des algorithmes (changer d'algorithme facilement, contrairement au RFC 2385, qui ne permettait que MD5) et celle des clés (changer de clés sans casser les sessions entre routeurs). À l'heure actuelle, c'est très loin d'être le cas.
— Une fois ces mécanismes en place, développer un KMP, un "*Key Management Protocol*" permettant la gestion et le remplacement automatique des clés. Ce dernier protocole devrait être aussi indépendant du protocole de routage que possible.

Pour que les mécanismes nouveaux aient des chances de succès, ils doivent pouvoir être déployés sans ajouter de complexité par rapport à ce que font déjà les opérateurs (qui connaissent SSH, TCP-MD5, HTTPS et les certificats, etc). Le but n'est pas de faire un système qui empêche l'opérateur de faire une erreur (comme d'utiliser « foobar » comme mot de passe pour tous les systèmes) mais de faire un système qui soit utilisé dans le monde réel, et pour cela, la simplicité et la déployabilité sont des critères essentiels (mais très souvent oubliés par les experts en sécurité, qui se soucient davantage de faire des systèmes parfaits que des systèmes déployables, cf. section 6).

Avec une gestion manuelle des clés, on ne peut gérer de manière raisonnablement sûre que des petits réseaux. La deuxième étape sera donc de déployer le mécanisme de gestion automatique.

Le travail d'amélioration de chaque protocole de routage est décrit en section 4.2 sous forme d'une liste de tâches :

— Analyser le protocole actuel pour déterminer de façon précise quels sont ses mécanismes de sécurité présents,
— Déterminer ce qui lui manque pour atteindre le premier état de sécurité souhaité (agilité des algorithmes et possibilité de changer de clés en vol),
— Analyse des étapes nécessaires pour aller du premier point au deuxième sans tout casser, en faisant particulièrement attention aux questions de déployabilité dans le monde existant,
— Analyser le futur KMP (protocole de gestion de clés) pour ce protocole de routage particulier : a-t-il des demandes spécifiques ?
— Déterminer ce qui manque pour atteindre le deuxième état de sécurité (gestion automatique des clés),
— Normaliser l'adaptation du KMP à ce protocole et déployer le résultat.

On l'a vu en section 2, KARP classe les protocoles de routage en catégories selon leurs propriétés. La section 5 examine les points qui sont spécifiques à chaque catégorie. BGP, LDP et quelques autres sont dans la catégorie « messages un-vers-un et clés par pair ». BGP fonctionne toujours sur TCP, LDP parfois sur TCP et parfois sur UDP. Pour le cas de TCP, une bonne partie du travail a déjà été faite dans le groupe TCPM <http://tools.ietf.org/wg/tcpm>, avec la technique d'authentification AO (RFC 5925) qui a les propriétés voulues pour la première phase du travail de KARP (agilité des algorithmes et changement de clé facile). Il ne reste donc que le cas de LDP sur UDP.

Pour la catégorie « un-vers-plusieurs avec clés par groupe », qui comprend OSPF, IS-IS et RIP, rien de générique n'est fait. Le problème est ici bien plus difficile, d'autant plus que ces protocoles n'utilisent pas en général de protocole de transport standard, et ne peuvent donc pas réutiliser un mécanisme fourni par la couche 4 (comme peut le faire BGP avec AO). Ceci dit, le RFC 7166 fournit une méthode d'authentification pour OSPF qui, grâce à une indirection, fournit l'agilité de l'algorithme de chiffrement, qu'on peut changer en vol.

BFD est un cas à part et qui nécessitera sa propre équipe. Par exemple, il est beaucoup plus sensible aux délais que les autres. Pour lui, une milliseconde est très longue.

On l'a vu, ce RFC répète régulièrement qu'il est essentiel de prévoir un déploiement incrémental des nouveaux mécanismes de sécurité. Pas question d'ignorer l'existant. La section 6 insiste sur ce point. Contrairement à une attitude fréquente chez les experts en sécurité, qui est de chercher une solution parfaite, KARP va essayer de trouver des solutions qui puissent être déployées par étapes, sans casser le routage actuel, même si ces solutions ne sont pas les meilleures, question sécurité. Par exemple, les routeurs configurés avec les nouveaux mécanismes doivent pouvoir interagir sans ennuis avec les vieux routeurs non sécurisés.

Un des problèmes de sécurité les plus difficiles dans l'Internet est celui des attaques par déni de service (section 7). Ces attaques touchent aussi les routeurs et les protocoles de routage. Il ne faut donc pas que les nouvelles techniques conçues dans le cadre du groupe KARP aggravent le problème. Par exemple, les calculs cryptographiques peuvent être coûteux, surtout pour les ressources matérielles souvent limitées des routeurs, et les protocoles ne doivent donc pas permettre à un attaquant de faire faire au routeur une énorme quantité de ces calculs. Pour éviter cela, il faut permettre au routeur de faire des vérifications non-cryptographiques, donc bien plus légères, avant de se lancer dans les calculs compliqués. Par exemple, le RFC 5082 utilise le TTL du paquet entrant (quelque chose de trivial à vérifier) pour empêcher certaines attaques. Il est important, dans le cadre de KARP, de développer et de documenter de telles alternatives. D'autre part, le protocole doit être conçu de manière à ce que ce soit l'initiateur de la connexion (un attaquant potentiel) qui ait le plus de travail cryptographique à faire, et qui doive maintenir un état pendant ce temps.

La section 9 du RFC, la traditionnelle section sur la sécurité, détaille quelques points précis qui n'avaient pas trouvé leur place dans le reste du RFC. Par exemple, il est important de considérer aussi si le protocole de routage qu'on veut protéger est un IGP ou un EGP (certains protocoles peuvent être utilisés pour les deux, comme BGP avec son mode iBGP). Les deux sont sans doute aussi importants mais les menaces et les solutions ne seront pas forcément les mêmes. Un routeur purement interne et qui n'a aucun accès à l'Internet est ainsi sans doute moins menacé qu'un routeur BGP posé sur un point d'échange avec des dizaines d'autres routeurs inconnus.

Autre question transversale, celle des clés partagées ou uniques. Faut-il utiliser la même clé à plusieurs endroits ? Le débat est simple : c'est la sécurité (clés uniques !) contre la commodité (clés partagées). Actuellement, dans la grande majorité des environnements, les opérateurs ont choisi la commodité, réutilisant la même clé à plusieurs endroits. Les clés des routeurs sont stables dans l'espace

(utilisées dans plusieurs routeurs) et dans le temps (on les change rarement, voire jamais et certains routeurs gardent la même clé toute leur vie). L'un des buts de KARP est de casser ce dilemne « sécurité ou commodité » en fournissant des mécanismes de gestion de clés qui soient sûrs (clés uniques) tout en étant faciles à déployer.

Enfin, la section 9.4 discute du dernier problème transversal, la distribution des clés. La méthode la plus courante aujourd'hui est un mécanisme « hors-bande », par exemple l'administrateur qui se connecte en SSH au routeur et entre manuellement les clés dans la configuration. Ça ne passe pas tellement à l'échelle (il faudrait automatiser les connexions SSH, par exemple avec Capistrano, pour faire l'opération sur N routeurs). Et changer toutes les clés en cas, par exemple de départ d'un administrateur (ou, pire, en cas de compromission de la clé) est trop lent. L'approche d'un KMP, protocole de gestion de clés, est donc préférée par KARP, comme nous l'avons déjà vu. Mais le RFC ne cache pas que le KMP a ses propres problèmes : lenteur au début (davantage de calculs cryptographiques) et surtout risques liés au manque de maturité des programmes, les KMP étant encore chose récente.

Une des possibilités envisagées est de réutiliser un KMP existant comme IKE, adapté au monde du routage, mais tournant en dehors du protocole de routage. L'autre possibilité est un nouveau KMP, embarqué dans le protocole de routage. Mais la décision n'est pas encore prise.